



Protocolo Spanning Tree



Redes de comunicação

GRUPO DE REDES DE COMUNICAÇÃO

ISEL - DEETC

10/10/2014

ISEL- DEETC

1

Agenda



- Revisão de Tópicos Anteriores
 - Protocolo *Spanning Tree*
 - Exemplo Teórico
 - Exercício Laboratorial
-



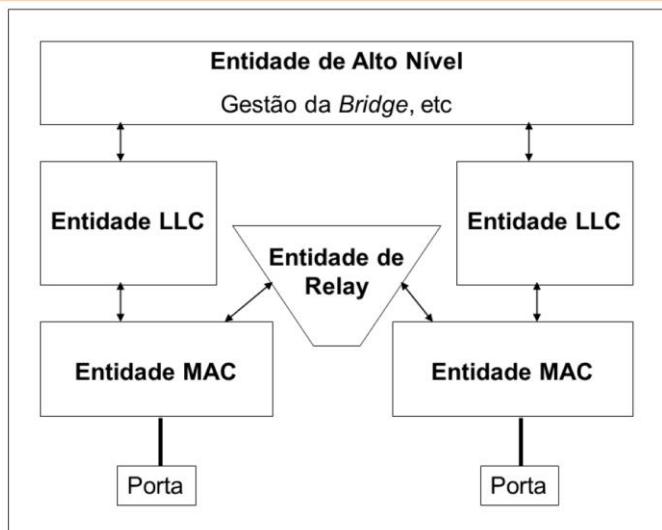
Funcionamento das *Bridges*

- As *Bridges/Switches* são dispositivos nível 2 (OSI) que realizam a comutação dos pacotes baseadas no endereço nível 2 (MAC) das máquinas
- As *Bridges* configuram-se de modo “transparente” para manter uma topologia de rede livre de ciclos
 - Apenas um caminho deve estar activo para qualquer destino na rede
- Caso existe alguma alteração na rede as *Bridges* reconfiguram-se
- As *Bridges* mantêm uma tabela de encaminhamento interno com informação da localização das máquinas que **vão aprendendo e actualizando**

- As *Bridges* são dispositivos nível 2 (OSI) que realizam a comutação das tramas baseadas no endereço nível 2 (MAC) das máquinas.
- As *Bridges* configuram-se de modo transparente para manter uma topologia de rede livre de ciclos. Entende-se como “transparente” o facto de as *Bridges* se configurarem de forma automática e sem ser necessário a sua configuração por parte de um administrador de rede.
 - Desta configuração resulta que apenas um caminho deve estar activo para qualquer destino na rede.
 - Caso existe alguma alteração na rede as *Bridges* reconfiguram-se de forma automática.
 - As *Bridges* mantêm uma tabela de encaminhamento interno com informação da localização das máquinas que aprendem e atualizam.



Arquitectura das Bridges



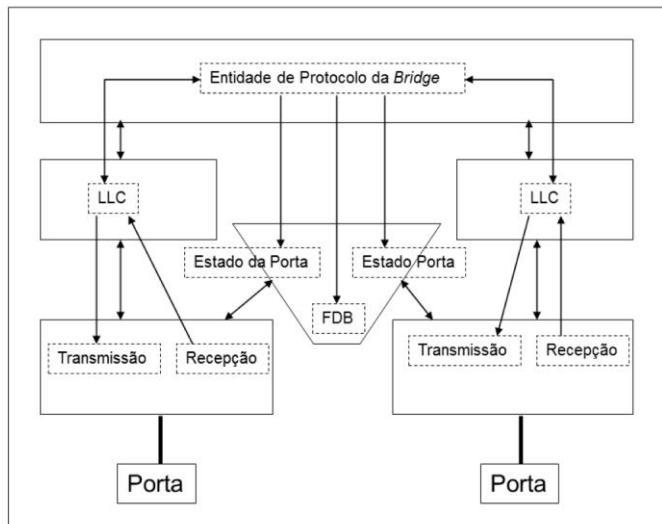
A arquitetura das Bridges é composta por diferentes níveis que permitem realizar a função de comutação das tramas entre as portas das Bridges. Cada porta de uma Bridge está ligada a uma única rede como se pode ver no exemplo apresentado com uma *Bridge* com 2 portas.

A entidade MAC tem as funções que permitem receber e enviar pacotes através das portas da Bridge.

A entidade de Relay permite realizar as funções de comutação das tramas entre as portas das Bridges.

A entidade de alto nível das Bridge é responsável pelas funções de gestão das *Bridges* e pelo cálculo da topologia de rede.

Fluxo de Controlo Numa Bridge

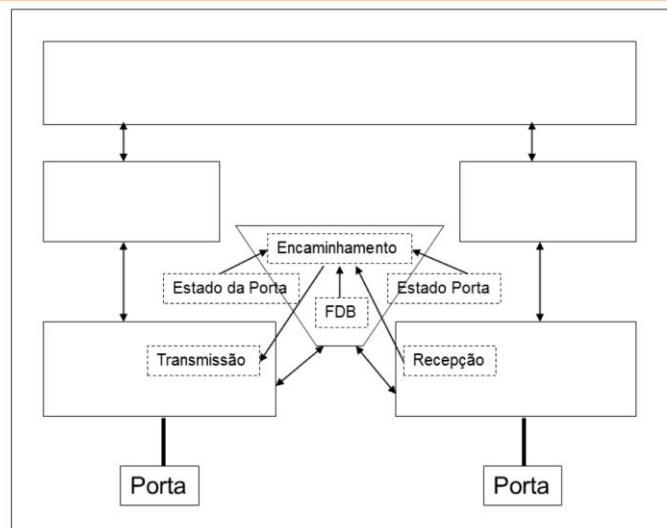


Existem dois tipos de fluxos nas *Bridges*, o fluxo de controlo que corresponde a uma troca de mensagens (BPDU) geradas pelas *Bridges* que permitem a sua configuração automática.

O percurso interno das mensagens de controlo, passa pelas entidades MAC das portas de entrada e saída e passa também pela entidade alto nível da *Bridge* que é a responsável por definir os estados em que as portas devem ficar.

A entidade de alto nível da *Bridge*, para além de definir o estado das portas da *Bridge*, gera a tabela interna de encaminhamento das *Bridges*.

Fluxo de Dados Numa Bridge



O fluxo de dados numa *Bridge* corresponde à comutação das tramas enviados pelos utilizadores da rede (PC dos Utilizadores, Servidores, etc).

A entidade de *Relay* é responsável por realizar a comutação das tramas tendo em conta o estado em que as portas se encontram e o conteúdo da tabela de comutação.

Desta forma as *Bridges* conseguem enviar as tramas apenas para as portas onde os destinatários se encontram se eles fizerem parte da tabela de comutação.

Modos de Comutação (Retransmissão)



- Modo 1 - *Cut-through*
 - Envia logo após endereço de destino disponível
- Modo 2 - *Modified Cut-through*
 - Envia logo após receber 64 bytes (dimensão mínima da trama) garante que não houve colisão na trama
- Modo 3 - *Store and Forward*
 - Envia após receber toda a trama e verificar que não tem erros garante que não houve erros nem colisão na trama



A estrutura do cabeçalho de uma trama nível 2 (MAC) é composto por um preâmbulo com 8 octetos, seguido da informação de endereço destino e endereço origem e tipo/dimensão do pacote. Seguem-se os campos de dados do pacote e o campo de verificação de erros (FCS).

A comutação ou retransmissão de tramas nas *Bridges* pode ser realizada de 3 formas diferentes.

No 1º Modo a *Bridge* inicia a retransmissão das tramas assim que é conhecido o endereço de destino dos pacotes para que possa ser consultada a tabela de comutação e reenviar para a porta de destino. Este modo é o mais rápido pois o atraso provocado é apenas o estritamente necessário. Tem a desvantagem de não haver validações das tramas.

No 2º Modo a *Bridge* inicia a retransmissão das tramas assim que recebe uma quantidade de octetos (64) que garanta que a trama recebida não tem posteriormente colisão. Este modo embora tenha um tempo de atraso um pouco maior, garante que a trama recebida não irá ter colisões pelo que não retransmite tramas com colisão.

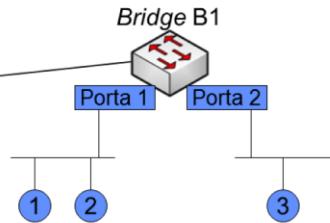
No 3º Modo a *Bridge* só inicia a retransmissão das tramas após receber toda a informação da trama e confirmar pelo campo de verificação de erros que a trama não tem erros. Este modo é o que provoca maior tempo de espera, no entanto garante que as tramas reenviadas não têm erros e não sofreram colisão.

Tabela de Comutação das Bridges



Tabela de Comutação da Bridge B1
(Forward Database – FDB)

Endereço	Porta	Tempo
1	1	
2	1	
3	2	



Para realizar a comutação das tramas, as *Bridges* têm uma tabela interna onde armazenam os endereços MAC das máquinas e a porta por onde entraram as tramas enviadas por essas máquinas.

É com base nesta tabela que depois a *Bridge*, face a um endereço de destino, se o endereço for conhecido a *Bridge* reenvia a trama para a porta respetiva, e se não for conhecido envia-a para todas as portas.

No exemplo apresentado, pode-se ver uma *Bridge B1* com duas portas (porta 1 e porta 2) em que na porta 1 liga uma rede que tem ligadas duas máquinas a máquina 1 e 2. E na porta 2 está ligada a máquina 3.

Após serem trocadas tramas que permitam à *Bridge* atualizar os endereços da tabela de comutação, o resultado será o apresentado na tabela (FDB).

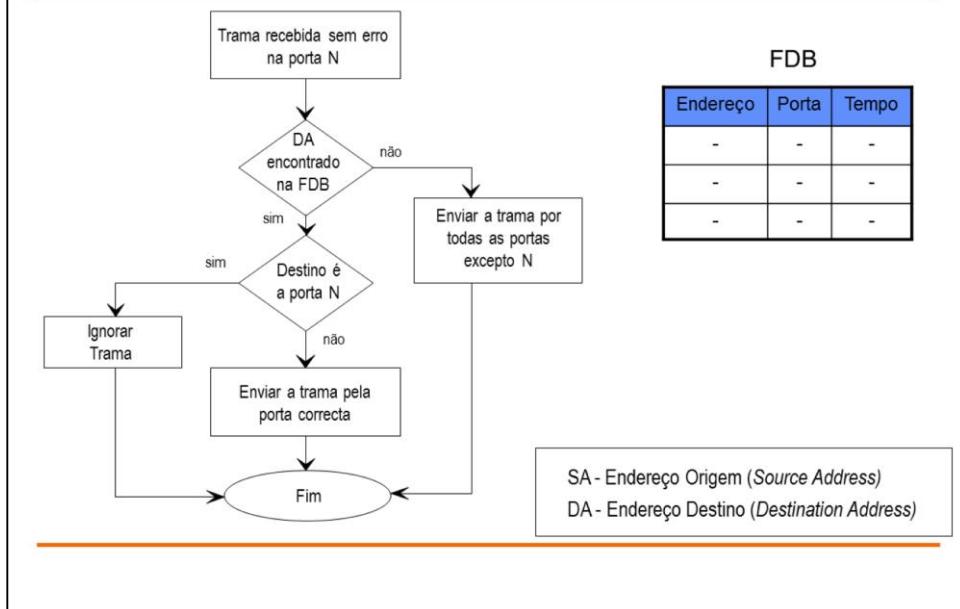
A coluna endereço contém os endereços MAC das máquinas.

A coluna porta contém o número da porta por onde entrou o pacote com o endereço correspondente.

A coluna Tempo permite definir um limite de tempo de armazenamento dos endereços na tabela.



Processo de Comutação



A forma como as Bridges realizam a comutação é realizada da seguinte forma:

Se a trama for recebida sem erros na porta N, a *Bridge* vai procurar se o endereço de destino da trama se encontra na tabela de encaminhamento (Condição 1).

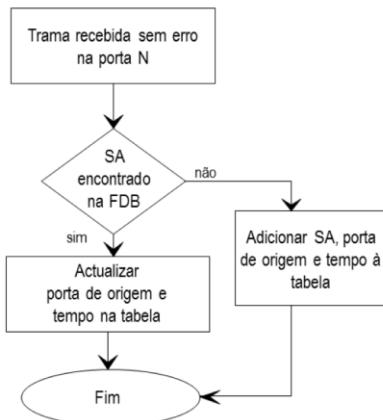
(Condição 1 - Sim) - Se o endereço de destino se encontra na tabela de comutação vai verificar se o número da porta que se encontra na tabela é igual ao número da porta por onde a trama entrou (Condição 2).

(Condição 2 - Não) - Se o número da porta da tabela for diferente do número da porta por onde entrou a trama então a trama é encaminhada pela porta que está indicada na tabela.

(Condição 2 - Sim) - Se o número da porta da tabela for igual ao número da porta por onde entrou então a trama é ignorada. Nesta situação a *Bridge* assume que a máquina recebeu também a trama.

(Condição 1 - Não) - Se o endereço de destino não se encontra na tabela de comutação então envia por todas as portas exceto a porta por onde entrou a trama.

Processo de Aprendizagem



Endereço	Porta	Tempo
-	-	-
-	-	-
-	-	-

SA - Endereço Origem (Source Address)
 DA - Endereço Destino (Destination Address)

A forma como as Bridges aprendem os endereços MAC e as portas de entrada é feita da seguinte forma:

Se a trama for recebida sem erros na porta N a *Bridge* vai verificar se o endereço de origem se encontra na tabela de comutação (Condição 1).

(Condição 1 - Não) - Se o endereço de origem ainda não existe na tabela de comutação, então é adicionada uma entrada na tabela de comutação com a informação do endereço MAC da máquina, porta por onde entrou e é iniciado o contador de tempo de armazenamento.

(Condição 1 - Sim) - Se o endereço de origem já existir na tabela de comutação, então é atualizada a informação relativa à porta de entrada e reiniciado o contador de tempo de armazenamento.

Tempos nas Tabelas de Comutação



- Os endereços aprendidos e colocados na tabela de comutação têm uma duração limitada
- O valor de tempo usado para esse efeito designa-se por *Ageing Time* e pode conter valores entre 10 segundos e 1 milhão de segundos
- O valor de omissão desta duração é 300 segundos
- O objectivo deste valor é garantir que se um terminal mudar de localização ou se deixar de estar ligado o seu endereço MAC não fica para sempre nas tabelas de comutação das *Bridges*

Os endereços aprendidos e colocados na tabela de comutação têm uma duração limitada.

O valor de tempo usado para esse efeito designa-se por *Ageing Time* e pode conter valores entre 10 segundos e 1 milhão de segundos.

O valor de omissão desta duração é 300 segundos.

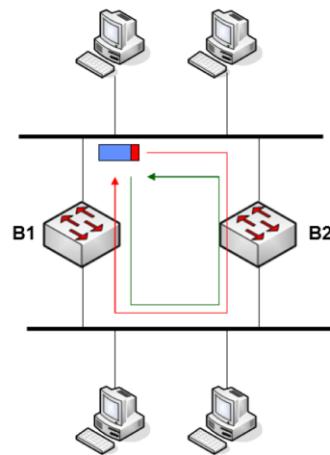
Quando o tempo de duração expira o endereço bem como a respetiva informação são removidos da tabela de comutação.

O objetivo deste valor é garantir que, se um terminal mudar de localização ou se deixar de estar ligado o seu endereço MAC, não fica para sempre nas tabelas de comutação das *Bridges*.

Problemas de Ciclos nas Redes



- Tramas de Broadcast - Os ciclos nas redes originam que as tramas de broadcast circulem na rede indefinidamente em ambas as direções
- Tramas Unicast - Os ciclos na rede podem originar que tramas unicast também circulem na rede indefinidamente



A existência de ciclos nas redes originam a que as tramas nível 2 possam chegar ao mesmo destino por diferentes caminhos.

Neste caso como se pode ver na figura, quando é enviada uma trama de um dos terminais na rede, existe o sentido de circulação assinalado a vermelho e o sentido de circulação assinalado a verde.

As tramas podem ter destinatários um único terminal (*Unicast*) ou para todos os terminais (*Broadcast*).

Nas tramas Broadcast o facto de não ter um destinatário identificado, pelo encaminhamento que já vimos anteriormente, as tramas são encaminhados por todas as portas exceto pela porta por onde entrou. Assim, se um terminal enviar uma trama de broadcast, este vai circular indefinidamente na rede em ambos os sentidos.

Para além das tramas de Broadcast originarem o problema de entrada em ciclo, as tramas com um único destinatário (*Unicast*) também podem sofrer do mesmo problema. Se for enviado uma trama com um endereço desconhecido na rede, ou de uma máquina que se desligou e entretanto deixou de constar das tabelas de comutação também vai ficar em ciclo indefinidamente.

Em ambas as situações, o facto das tramas circularem indefinidamente na rede em ambas as direções originam a que as *Bridges* recebam a mesma trama em ambas as portas. Cada vez que a trama entra por uma porta diferente a *Bridge* atualiza a sua informação na tabela de comutação o que provoca uma alternância entre as portas de entrada.

Protocolo *Spanning Tree*



- Protocolo *Spanning Tree*
 - MAC Bridge Spanning Tree Protocol (STP)(802.1D)
 - Publicação original pelo IEEE 802.1D-1990
 - Revisão versão IEEE 802.1D-1998
 - Revisão versão IEEE 802.1D-2004
- Protocolos Relacionados
 - 802.1w – *Rapid Spanning Tree Protocol* (RSTP)
 - 802.1s – *Multiple Spanning Tree Protocol* (MSTP)

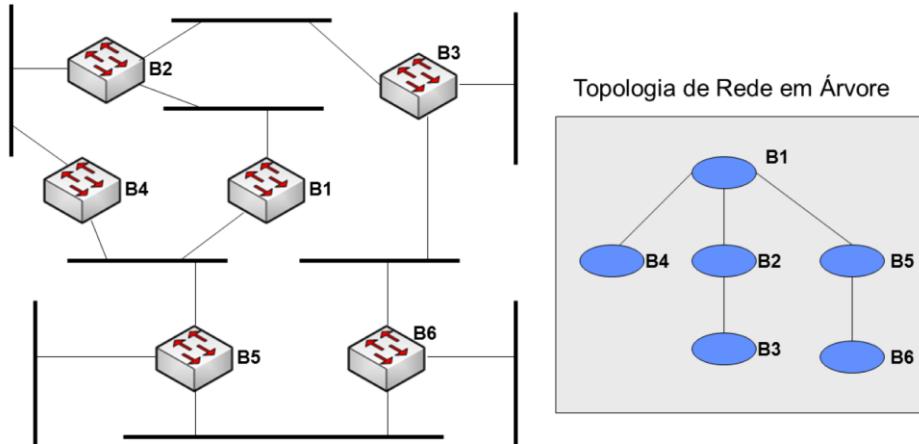
O protocolo Spanning Tree foi publicado originalmente pela norma IEEE 802.1D em 1990. Esta norma aplica-se a redes locais e metropolitanas para *Bridges* com encaminhamento MAC.

Mais tarde foram introduzidas correções técnicas e editoriais e foram adicionados extensões à norma.

Mais recentemente em 2004 foram introduzidas mais correções técnicas e alterações à norma tendo sido adicionados extensões como são exemplo a norma 802.1w (Rapid Spanning Tree Protocol) e a relação entre a norma 802.1D e a norma 802.1Q (definição das VLAN).

Os protocolos Rapid Spanning Tree Protocol e Multiple Spanning Tree Protocol são evolução do protocolo original Spanning Tree e permitem resolver as limitações deste protocolo, nomeadamente, tempo de convergência e uma topologia de rede por cada VLAN existente.

Exemplo de Aplicação do Protocolo



Em termos gerais, se tivermos uma rede como é apresentada na figura, podemos ver que existem vários ciclos na rede.

Como foi apresentado anteriormente, estes ciclos originam os problemas apresentados.

O objectivo principal, é as *Bridges* trocarem informação entre elas para detetarem os ciclos existentes na rede e permitirem configurar uma topologia lógica de rede sem ciclos.

A figura de Topologia, mostra uma topologia lógica possível de organização das *Bridges* para removerem os ciclos.

Desta forma, as *Bridges* vão atuar nas portas que ligam às redes para que apenas exista um caminho possível entre cada dois troços de rede.

É através da topologia definida que as *Bridges* vão trocar informação para garantir que a topologia se mantém e não existem alterações na rede.

No caso de haver alterações as *Bridges* vão definir uma nova topologia.

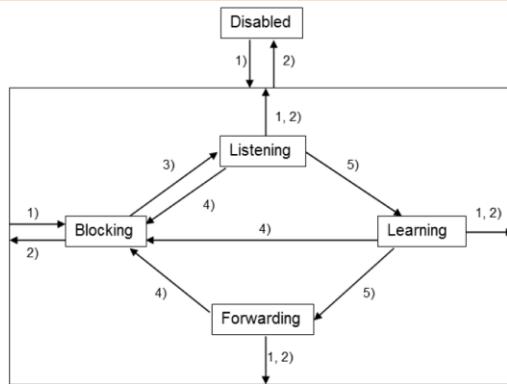
Características do Protocolo



- As *Bridges* transmitem mensagens entre elas que permitem calcular a topologia em árvore
- As mensagens de controlo trocados entre as *Bridges* designam-se por *Bridge Protocol Data Units (BPDU)*
- As portas têm diferentes estados possíveis
 - *Blocking* – Descarta as tramas de dados que recebe
 - *Listening* – Construção da topologia ativa
 - *Learning* – Preenchimento da tabela comutação
 - *Forwarding* – Envio / Recepção de pacotes
 - *Disabled* – Administrativamente desligada

- As *Bridges* transmitem mensagens entre elas que permitem calcular a topologia activa da rede em árvore
- Os pacotes trocados entre as *Bridges* designam-se por *Brigde Protocol Data Units (BPDU)*
- As portas têm diferentes estados possíveis
 - Blocking – Descarta as tramas que recebe
 - Listening – Construção da topologia ativa
 - Learning – Preenchimento da tabela encaminhamento
 - Forwarding – Envio / Recepção de pacotes
 - Disabled – Administrativamente desligada

Estados das Portas e suas Transições



- 1) Activação por gestão ou por inicialização
- 2) Desactivação por gestão ou falha
- 3) Processo de aprendizagem do algoritmo
- 4) Bloqueio pelo algoritmo
- 5) Transição por expiração de tempo de aprendizagem

Os Estados das portas das *Bridges* podem estar num de 5 estados (*Disabled*, *Blocking*, *Listening*, *Learning*, *Forwarding*).

A transição entre os estados pode ter várias origens:

- Pode ser devido a ações de gestão pelo administrador da rede.
- Pode ser devido a falhas na rede e os portos transitam de estado.
- Pode ser devido ao protocolo de Spanning Tree para controlar os ciclos na rede.

As transições para o estado *Disabled* (porta desativa) pode ter origem em ações de gestão ou falhas da porta (Ações 1 e 2).

As transições para os estados *Blocking*, *Listening*, são definidas pelo algoritmo de aprendizagem (Ações 3 e 4).

As transições para os estados *Learning* e *Forwarding* são desencadeados por fim de contagens de tempo de relógios internos existentes nas *Bridges* (Ação 5). Esta ação garante que o processo de definição da topologia de rede tem um tempo definido e consistente entre as *Bridges*.

Processamento dos pacotes



- Processamento dos pacotes nos diferentes estados:
- *Blocking* – As tramas de dados recebidas são descartadas, os BPDU não são
- *Listening* – As tramas de dados recebidas são descartadas mas as portas encontram-se no processo de transição para o estado *Learning*
- *Learning* – As tramas de dados recebidas não são reenviadas, mas os endereços e as portas são aprendidos e colocados nas tabelas de comutação
- *Forwarding* – As tramas de dados recebidas são processadas e os endereços origem são aprendidos e são reenviadas
- *Disabled* – A porta está desativada

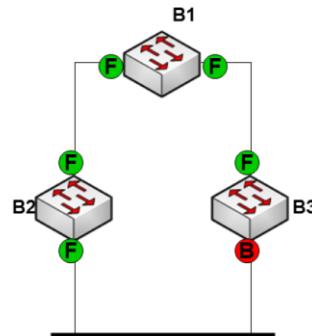
O processamento das tramas nos diferentes estados é realizado da seguinte forma:

- No estado Blocking as tramas recebidas são descartadas.
- No estado Listening as tramas recebidas são descartadas mas as portas encontram-se no processo de transição para o estado Learning.
- No estado Learning as tramas recebidas são descartadas, mas os endereços e as portas são colocados nas tabelas de encaminhamento.
- No estado Forwarding as tramas recebidas são processadas e os endereços origem são aprendidos.
- No estado Disabled a porta está desativada.

Estados *Forward* e *Blocked*



- Para realizar comutação de tramas as portas têm de estar no estado *Forward*
- Para prevenir os ciclos as portas ficam no estado *Blocking* e não reenviam as tramas recebidas
- Os estados *Listening* e *Learning* são usados para cálculo da topologia de rede
- No estado *Disabled* não existe troca de tramas



As portas que estão no estado Forward ou são Root Ports ou Designated Ports. Todas as outras portas são colocadas no estado Blocking

- Para realizar comutação de tramas as portas têm de estar no estado Forward.
- Para prevenir os ciclos as portas ficam no estado Blocking e não reenviam as tramas recebidos.
- Os estados Listening e Learning são usados para cálculo da topologia de rede.
- No estado Disabled não existe troca de pacotes.

Na figura é apresentado uma rede com ciclo onde se pode ver um exemplo de topologia de rede que retira o ciclo.

Neste exemplo as *Bridges* 1 e 2 têm todos os seus portos no estado Forward (assinalados com um círculo verde com a letra F) o que permite processar as tramas de dados recebidas.

A Bridge 3 tem um porto no estado Forward e outro no estado Blocked (assinalado com um círculo vermelho com a letra B).

As portas que estão no estado Forward, ou são Root Ports, ou são Designated Ports (como veremos mais à frente), todas as outras portas são colocadas pelo algoritmo no estado Blocking.

Do ponto de vista lógico, como se pode ver na figura, a rede funciona como se tivesse a porta bloqueada desligada porque não reenvia as tramas recebidas.

Troca de Mensagens BPDU



- Troca de informação entre as *Bridges*
 - As mensagens trocadas entre as *Bridges* são do tipo BPDU
 - As mensagens BPDU de configuração (C-BPDU – 35 octetos) são enviadas pela *Root Bridge* para as restantes *Bridges*
 - As mensagens de mudança de topologia (TC-BPDU – 4 octetos) são enviadas pelas *Bridges* para recalcular a topologia activa na sequência de uma alteração na rede

Exemplo de um C-BPDU

<i>Protocol Identifier</i>	2
<i>Protocol Version Identifier</i>	1
<i>BPDU Type</i>	1
<i>Flags (TCA, Reserved, TC)</i>	1
<i>Root Identifier</i>	8
<i>Root Path Cost</i>	4
<i>Bridge Identifier</i>	8
<i>Port Identifier</i>	2
<i>Message Age</i>	2
<i>Max Age</i>	2
<i>Hello Time</i>	2
<i>Forward Delay</i>	2

A Troca de informação entre as *Bridges* é feita com base em mensagens BPDU trocadas entre elas.

As mensagens que permitem configurar e calcular uma topologia de rede são as mensagens BPDU de configuração (C-BPDU – compostas por 35 octetos) são enviadas pela *Root Bridge* para as restantes *Bridges*. Todas as outras *Bridges* reenviam as mensagens enviadas pela *Root Bridge*.

Quando é detetada uma mudança na rede, as *Bridges* enviam as mensagens de mudança de topologia (TC-BPDU – composta por 4 octetos). Estas mensagens são enviadas pelas *Bridges* para informar a *Root Bridge* para recalcular a topologia activa na sequência de uma alteração na rede.

Campos da Mensagem C-BPDU



- Protocol ID = 0x0000
- Protocol Version ID = 0x00
- Protocol Type = 0x00
- Root Identifier = ID da *Root Bridge*
- Root Path Cost = Custo do caminho até à *Root Bridge* (é armazenado o melhor caminho)
- Bridge Identifier = ID da *Bridge* que envia o BPDU
- Port Identifier = ID do porto por onde sai o BPDU

Os valores definidos para o identificador do protocolo, versão e tipo são 0x00. O identificador Root é o da Bridge assumida como sendo a Root Bridge. O identificador da Bridge, é o da Bridge que está a enviar o pacote. O identificador do Porto é o identificador do porto por onde sai o BPDU. O custo de caminho, é calculado à medida que o pacote vai circulando pelas Bridges e contém o custo entre a Bridge que está a enviar e a Root Bridge. O objectivo é encontrar o menor custo de caminho entre qualquer Bridge e a Root Bridge.

Campos da Mensagem C-BPDU



- Message Age = (Na *Root Bridge*=0, nas outras tem um incremento correspondente ao número de *Bridges* por onde passou)
- Max Age = Tempo de armazenamento dos BPDU
- Hello Time = Tempo entre cada envio de BPDU
- Forward Delay = Tempo de cálculo da topologia de rede

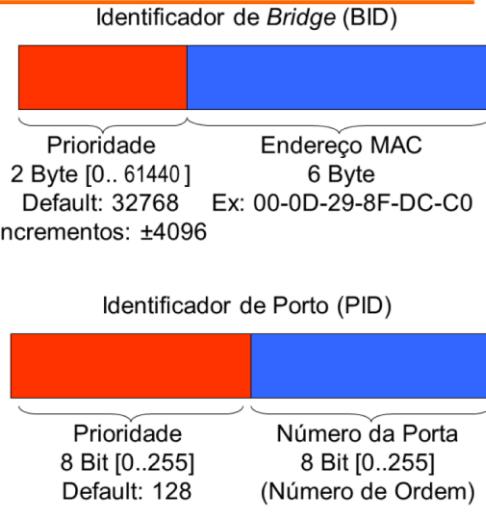
O valor de Message Age permite saber quantas *Bridges* percorreu o pacote. Na *Root Bridge* o BPDU sai com valor = 0 e vai sendo incrementado à medida que vai ser retransmitido pelas *Bridges*.

Os valores de Max Age, Hello Time e Forward Delay, permitem às *Bridges* conhecer e sincronizar os tempos que estão configurados na *Root Bridge*. Com estes valores as *Bridges* configuram-se consistentemente.



Identificadores Bridge / Porto

- Identificador de *Bridge* mais prioritário:
 - 1º - A *Bridge* com menor valor de prioridade;
 - 2º - Em caso de valores de prioridade iguais, menor valor de endereço MAC
- Identificador de *Porto* mais prioritário:
 - 1º - A porta com menor valor de prioridade;
 - 2º - Em caso de valores de prioridade iguais, menor valor número de ordem da porta.



Os identificadores das Bridges são compostos por dois valores: O valor de prioridade da Bridge; e o valor de endereço MAC da Bridge.

O valor de prioridade é atribuído por ações de gestão com valores entre 0 e 61440 sendo o valor omissão igual a 32768, em incrementos de 4096.

O valor de endereço MAC é constituído por 6 Bytes e corresponde ao endereço MAC da Bridge.

Um identificador de *Bridge* é mais prioritário que outro se:

- 1º - O valor de prioridade da *Bridge* tem menor valor de prioridade;
- 2º - Em caso de valores de prioridade iguais, o menor valor de endereço MAC é o identificador mais prioritário.

Os identificadores das portas das Bridges são compostos por dois valores: O valor de prioridade da porta; e o valor de número de ordem da porta.

O valor de prioridade é atribuído por acções de gestão com valores entre 0 e 255 sendo o valor omissão igual a 128.

O número de ordem da porta é constituído por 1 octeto e corresponde ao número de ordem da porta com valores possíveis entre 0 e 255.

Um Identificador de Porta mais prioritário que outro se:

- 1º - O valor de prioridade da porta tem menor valor de prioridade;
- 2º - Em caso de valores de prioridade iguais, o menor valor de número de ordem da porta é a porta mais prioritária.

Distância - Custos dos Segmentos



- Inicialmente o custo original era calculado pela fórmula:

$$Custo = \frac{1000}{Velocidade[Mbps]}$$

- Com o aparecimento da rede Gigabit surgiu o problema de cálculo de valores superiores a 1000 Mbps porque o valor usado é um valor inteiro.
- Foi decidido, pelo IEEE, modificar o custo e usar uma escala não linear apresentada na tabela e que é usada actualmente.

Tabela de Custos dos Segmentos

Velocidade	Custo
4 Mbps	250
10 Mbps	100
16 Mbps	62
100 Mbps	19
1 Gbps	4
10 Gbps	2

O cálculo das distâncias (custo do caminho) entre as *Bridges* é calculado em função dos débitos dos troços existentes entre as *Bridges*.

Inicialmente quando a norma foi proposta, o cálculo era feito através da fórmula apresentada, 1000 a dividir pela velocidade em megabit por segundo. Se por exemplo tivéssemos uma velocidade de 10 megabit por segundo o custo seria 100. Ou seja um custo inversamente proporcional à velocidade o que provoca para velocidades menores custos maiores.

Com o aparecimento da rede Gigabit surgiu o problema de cálculo de valores superiores a 1000 Mbps porque o valor usado é um valor inteiro e a divisão por valores superiores a 1000 causa custos decimais.

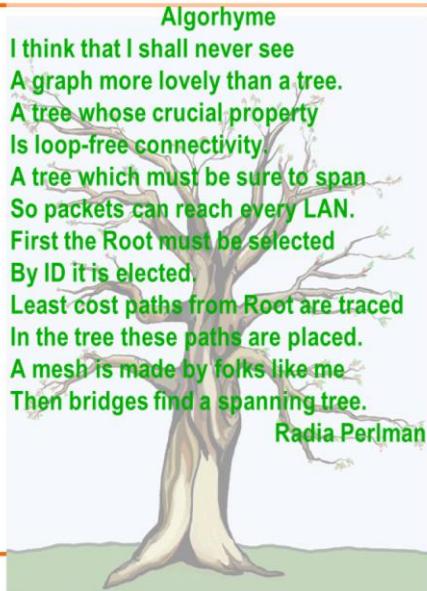
Foi decidido, pelo IEEE, modificar a fórmula de calculo dos custos e criou uma tabela com uma escala não linear apresentada e que é usada atualmente.

Por exemplo os valores de 10 Mbps, 100 Mbps e 1 Gbps têm respetivamente 100, 19 e 4 de custo.

Algoritmo Spanning Tree



- Algoritmo desenvolvido por Radia Perlman
- Integrado na especificação IEEE 802.1D-1990
- 1º é eleita a *Bridge Root Bridge* pelo seu *Bridge ID*
- 2º São eleitas as portas *Root Port* para cada *Bridge* que não é *Root Bridge*
- 3º São eleitas as portas *Designated Port* para cada segmento



Foi desenvolvido um Algoritmo desenvolvido por Radia Perlman para permitir realizar o cálculo da topologia de rede livre de ciclos.

Este algoritmo foi Integrado na especificação IEEE 802.1D em 1990.

Na sua proposta, a autora resumiu a ideia principal do algoritmo num poema que designou por “Algorhyme” e que é apresentado na sua forma original em Inglês.

Na introdução é apresentado a árvore como uma propriedade essencial para uma ligação em rede livre de ciclos.

Na segunda parte é introduzido o 1º passo do algoritmo que corresponde à eleição da *Bridge Root Bridge* pelo seu identificador.

Por fim é introduzido os dois últimos passos do algoritmo baseado no cálculo das menores distâncias entre as Bridges e a Root Bridge.

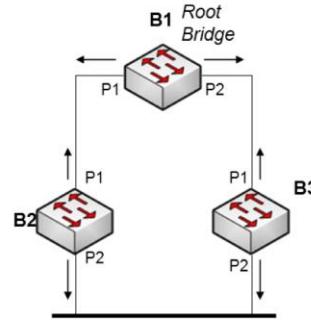
No 2º passo são eleitas as portas *Root Port* para cada *Bridge* que não é *Root Bridge*.

No 3º passo São eleitas as portas *Designated Port* para cada segmento.

Eleição da Root Bridge



- As Bridges enviam os BPDU com o seu BID assumindo-se como Root Bridge
- Todas as Bridges ficam a conhecer os valor de BID das outras Bridges
- A Bridge com o menor valor de BID é eleita como Root Bridge
- No exemplo apresentado a B1 é a Root Bridge



Identificadores das Bridges

B1: 32768-000D298FDCC0

B2: 32768-000D298FDCC1

B3: 32768-000D298FDCC2

O processo de eleição da Root Bridge é realizado da seguinte forma:

Inicialmente todas as Bridges enviam as mensagens BPDU com o seu identificador de Bridge e assumindo-se como Root Bridge.

Todas as Bridges recebem as mensagens com os identificadores das outras Bridges e apenas reenviam as mensagens cujo identificador é o menor identificador de todas as Bridges.

Ao fim de algum tempo, apenas a Root Bridge está a enviar as mensagens com o seu identificador como Root Bridge, e todas as outras estão a reenviar essas mensagens atualizando os seus campos.

Desta forma a Bridge com o menor valor de identificador é eleita como Root Bridge.

No exemplo apresentado, todas as Bridges têm o mesmo valor de prioridade (valor por omissão) e por isso o desempate é feito com base no endereço MAC.

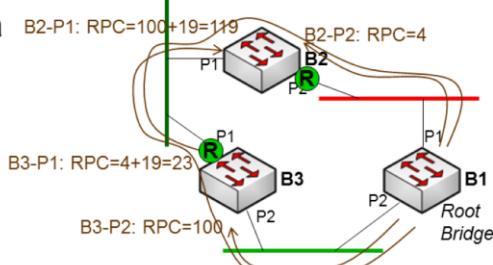
Como a Bridge B1 tem o menor valor de endereço MAC então a Bridge B1 é eleita como a Root Bridge.

Eleição dos Root Ports



- Cada Bridge calcula o valor de custo do caminho de cada porta até à Root Bridge (RPC)
- A porta Root Port (RP) de cada Bridge é a porta com menor custo

Porta	PC	RPC	RP
B1-P1	4	-	-
B1-P2	100	-	-
B2-P1	19	119	-
B2-P2	4	4	X
B3-P1	19	23	X
B3-P2	100	100	-



O RPC é incrementado na bridge destino

Velocidade	Custo
1 Gbps	4
100 Mbps	19
10 Mbps	100

RPC (Root Path Cost) – Custo do caminho até à Root Bridge
RP (Root Port) – Porta Root Port de cada Bridge

Todas as Bridges que não são Root Bridges calculam o valor de custo de caminho (RPC) de cada porta entre ela e a Root Bridge através das mensagens C-BPDU que vão chegando.

A porta da Bridge que tiver o menor valor de custo é eleita como Root Port da Bridge.

Cada Bridge que não é Root Bridge tem apenas uma porta definida como Root Port e com o respetivo custo de caminho até à Root Bridge.

No exemplo apresentado temos 3 Bridges com segmentos rede de diferentes velocidades.

O segmento a vermelho tem a velocidade de 1 Gbps que corresponde a um custo de 4. O segmento a verde tem a velocidade de 100 Mbps que corresponde a um custo de 19. O segmento a azul tem a velocidade de 10 Mbps que corresponde a um custo de 100.

Assumindo que a Bridge B1 é a Root Bridge, as outras duas Bridges B2 e B3 calculam os valores de RPC para as suas portas.

- No caso da porta 2 Bridge B2 o cálculo é feito somando o valor de custo do segmento a vermelho o que dá o valor de 4. No caso da porta 2 Bridge B3 o cálculo é feito somando o valor de custo do segmento a azul o que dá o valor de 100.

- No caso da porta 1 Bridge 3 o cálculo é feito somando o valor de custo do segmento a vermelho mais o custo do segmento a verde o que dá o valor de 4+19 e que corresponde a 23. No caso da porta 1 Bridge B2 o cálculo é feito somando o valor de custo do segmento a azul mais o custo do segmento a verde o que dá o valor de 100+19 e que corresponde a 119.

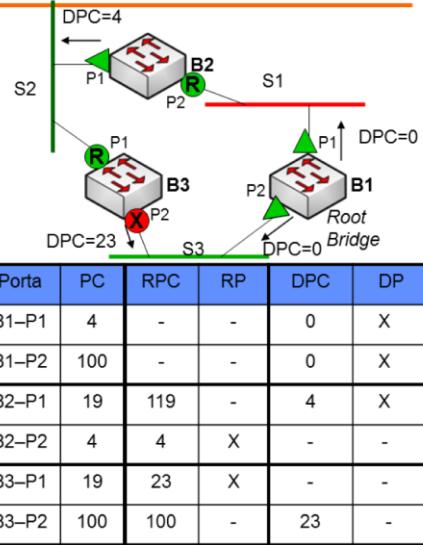
- Depois de calcular todos os custos RPC de todas as portas, cada Bridge vai decidir que a Root Port é a que tiver o menor valor. No caso da Bridge B2 a porta Root Port é a porta 2 com custo RPC=4. No caso da Bridge B3 a porta Root Port é a porta 1 com custo RPC=23.

Os valores de cálculo são resumidos na tabela apresentada (RPC) e são assinaladas as portas definidas como Root Ports (RP), na tabela com um X e na figura com um círculo verde na porta correspondente.

Eleição dos *Designated Ports*



- Cada segmento tem associado uma porta *Designated Port* de uma *Bridge* ligada a ele
- O custo de saída DPC das portas que não são *Root Port* é o valor de entrada RPC da *Root Port*
- É *Designated Port* de um segmento a porta com menor custo que está ligada a esse segmento
- O custo até à *Root* é do ponto de vista dos segmentos e não das *bridges*



Cada segmento de rede tem associado uma porta de uma *Bridge* ligada a ele que é considerada *Designated Port* do segmento.

Cada porta ligada ao segmento tem um valor de custo de saída DPC (o valor de custo de saída DPC da porta, corresponde ao valor RPC da porta *Root Port* da respectiva *Bridge*).

A porta ligada ao segmento que tiver menor custo de saída é eleita como *Designated Port* desse segmento.

No exemplo apresentado, assumindo que a *Bridge* B1 é a *Root Bridge* e os valores RPC são os calculados anteriormente, o cálculo dos valores DPC das portas é feito da seguinte forma:

- Para cada *Bridge* que não é *Root Bridge*. O custo DPC das portas que não são *Root Ports*, é o valor RPC da porta que é a *Root Port*. Este custo corresponde ao valor de custo de saída dos BPDU que vêm da *Root Bridge* e reflete a distância entre as *Bridges*.
- Desta forma, para o segmento S1 a vermelho existem duas portas ligadas a ele. A porta B1-P1 e a porta B2-P2. O custo DPC da porta B1-P1 por ser *Root Bridge* é igual a 0. A porta B2-P2 como é *Root Port* não é *Designated Port*. Assim a porta *Designated Port* do segmento S1 fica a porta B1-P1.
- Para o segmento S2 a verde existem duas portas ligadas a ele. A porta B2-P1 e a porta B3-P1. O custo DPC da porta B2-P1 é igual a 4. A porta B3-P1 como é *Root Port* não é *Designated Port*. Assim a porta *Designated Port* do segmento S2 fica a porta B2-P1.
- Para o segmento S3 a azul existem duas portas ligadas a ele. A porta B1-P2 e a porta B3-P2. O custo DPC da porta B1-P2 por ser *Root Bridge* é igual a 0. O custo DPC da porta B3-P2 é igual a 23. Assim o menor valor de DPC é o 0 sendo a porta *Designated Port* do segmento S3 a porta B1-P2.

Neste caso como a porta B3-P2 não é *Root Port* nem *Designated Port* é colocada no estado *Blocked*.

As portas *Designated Ports* são assinaladas com o Triângulo verde na figura, e as portas *Blocked* são assinaladas com um círculo vermelho com a letra X.

A tabela apresentada resume os cálculos realizados e assinala as portas *Designated Port* (DP) com um X.

Decisão sobre o “Melhor” BPDU



- Dados dois BPDU de configuração a escolha do melhor é feita pela seguinte ordem:
 - Se o Identificador da *Root Bridge* de um BPDU for menor que o de outro BPDU o valor menor é o melhor (maior prioridade)
 - Se os identificadores de *Root Bridge* forem iguais, compara o valor dos custos dos BPDU (RPC). O BPDU com menor valor de custo é o melhor.
 - Se os valores de Identificador de *Root Bridge* e custo forem iguais, verifica qual o menor valor de identificador de *Bridge*
 - Se todos os valores forem iguais o desempate é feito pelo menor valor de identificador da porta de saída
 - Se todos os valores anteriores forem iguais desempata o valor da porta de entrada (caso de um *switch -> hub -> duas portas de outro switch*)

Para decidir qual o melhor de 2 BPDU são usados os seguintes critérios de desempate e pela ordem apresentada:

- Se o Identificador da *Root Bridge* de um BPDU for menor que o outro BPDU o valor menor é o melhor.
- Se os identificadores de *Root Bridge* forem iguais compara o valor dos custos dos BPDU's. O BPDU com menor valor de custo é o melhor.
- Se os valores de Identificador de *Root Bridge* e custo forem iguais, verifica qual o menor valor de identificador de *Bridge*.
- Se todos os valores forem iguais o desempate é feito pelo menor valor de identificador da porta.

Tempos de cálculo da topologia



- A construção da árvore é controlada por 3 relógios
 - O relógio *Hello Time* – Tempo entre cada envio dos BPDU de configuração (valor típico 2 segundos)
 - O relógio *Forward Delay* – Duração da permanência nos estados *Listening* e *Learning* (valor típico de 15 segundos, cada)
 - O relógio *Max-Age* – Tempo durante o qual os BPDU são armazenados (valor típico de 20 segundos)

A construção da árvore é controlada por 3 relógios que são enviados nas mensagens BPDU:

O relógio ‘Hello Time’ corresponde ao tempo entre cada envio de BPDU de configuração (valor típico 2 segundos)

O relógio Forward Delay corresponde à duração dos estados Listening e Learning (valor típico de 15 segundos)

O relógio Max-Age corresponde ao tempo em que os BPDU são armazenados (valor típico de 20 segundos)

Utilização dos Relógios

O relógio Hello Time controla o espaço entre os C-BPDU enviados pela Root Bridge.

O relógio Forward Delay permite à Root Bridge controlar a passagem ao estado *forward* de modo consistente. É calculado em função da distância máxima numa rede entre 2 Bridges. O valor omissão foi pensado para 7 Bridges.

O relógio Max-Age foi pensado para que no caso uma *Bridge* falhar haja um novo processo de cálculo após cessar esse tempo.

Tempos de cálculo da topologia



- Utilização dos Relógios
 - O relógio *Hello Time* controla o espaço entre C-BPDU enviados pela *Root Bridge*
 - O relógio *Forward Delay* permite à *Root Bridge* controlar a passagem ao estado *forward* de modo consistente. É calculado em função da distância máxima numa rede entre 2 Bridges. O valor omitido foi pensado para um diâmetro máximo de 7 bridges
 - O relógio *Max-Age* foi pensado para que no caso uma *Bridge* falhar haja um novo processo de cálculo após cessar esse tempo

Detecção de Mudança na Rede

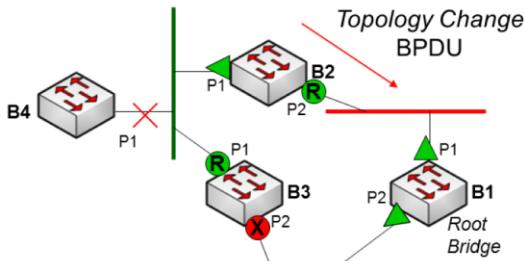


- Quando uma *Bridge* detecta mudança na rede inicia o processo de alteração
- Se for *Root Bridge* muda a *Flag* de *Topology Change* para verdade e inicia o contador de tempo de mudança de topologia
- Se não for *Root Bridge* e a *Flag* de *Topology Change* for falso então envia o TC-BPDU pela *root port* para a avisar a *Root Bridge*

Quando uma *Bridge* detecta mudança na rede inicia o processo de alteração:
Se for *Root Bridge* muda a *Flag* de *Topology Change* para verdade e inicia o contador de tempo de mudança de topologia.
Se não for *Root Bridge* e a *Flag* de *Topology Change* for falso então envia o TC-BPDU para a avisar a *Root Bridge*.



Detectção de Alteração na Rede



- A ligação da *Bridge B4* à rede quebrou-se
- Se a *Bridge B2* detetar a alteração de topologia e envia um BPDU de alteração de topologia pelo Root Port (caminho mais curto até à *Root Bridge*)
- As *Bridges intermédias* respondem com *Topology Change Acknowlege* e enviam o TC-BPDU pelo *Root Port*
- Quando a *Root Bridge* recebe o TC-BPDU envia os BPDU de configuração para recalcular a topologia activa

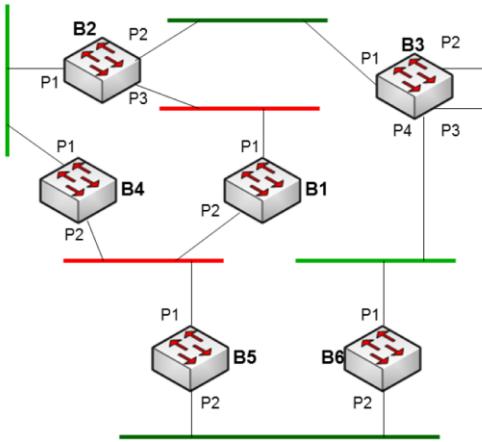
No exemplo apresentado e considerando que a Ligação da *Bridge B4* à rede quebrou-se:

- A *Bridge B2* detecta a alteração de topologia e envia um BPDU de alteração de topologia pelo Root Port (caminho mais curto até à *Root Bridge*).

As *Bridges intermédias* respondem com *Topology Change Acknowlege* e enviam o TC-BPDU pelo *Root Port*.

Quando a *Root Bridge* recebe o TC-BPDU envia os BPDU de configuração para recalcular a topologia activa.

Exemplo de Cálculo do Spanning Tree



Identificadores das Bridges

B1)	32768:00-0D-29-8F-DC-C1
B2)	32768:00-0D-29-8F-DC-C2
B3)	32768:00-0D-29-8F-DC-C3
B4)	32768:00-0D-29-8F-DC-C4
B5)	32768:00-0D-29-8F-DC-C5
B6)	32768:00-0D-29-8F-DC-C6

	Velocidade	Custo
—	1 Gbps	4
—	100 Mbps	19
—	10 Mbps	100

Todas as portas das Bridges têm a prioridade por omissão

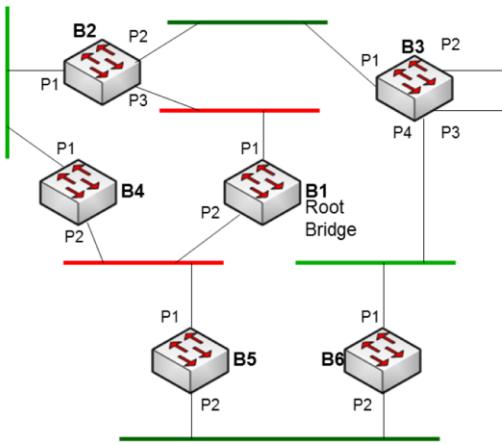
O exemplo teórico mostra uma topologia composta por 6 Bridges que estão interligadas por troços com diferentes velocidades (1Gbps, 100Mbps e 10Mbps).

A rede apresentada é composta por ciclos e o objectivo é calcular a topologia final da rede depois de aplicar o algoritmo de Spanning Tree.

Os identificadores das Bridges necessários ao Algoritmo são descritos na legenda.

A prioridade de todas as portas das Bridges é a prioridade de omissão.

Eleição da Root Bridge



	Velocidade	Custo
—	1 Gbps	4
—	100 Mbps	19
—	10 Mbps	100

1º Passo – Cálculo da *Root Bridge*.

A Bridge 1 (B1) é a *Root Bridge* dado possuir o menor rootID

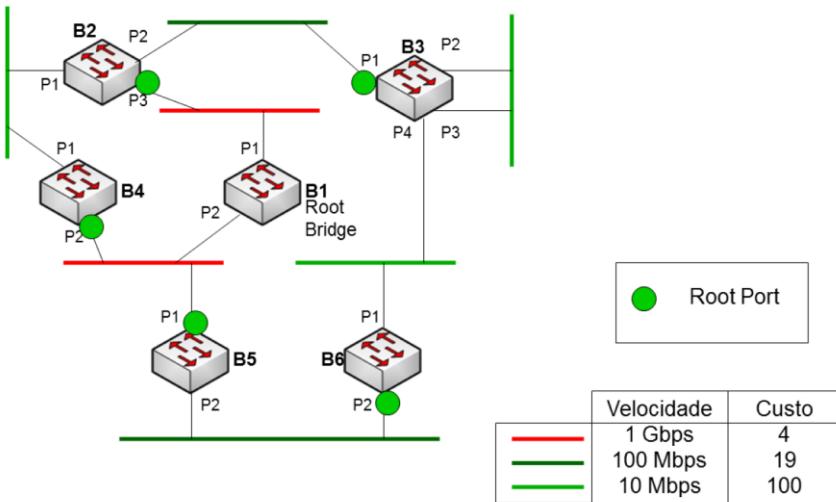


Valores de Cálculo de Root Path Cost

Porta	PC	RPC	RP	DPC	DP	Block
B1-P1	4	-	-			
B1-P2	4	-	-			
B2-P1	100	4+100=104	-			
B2-P2	19	4+19+100=123	-			
B2-P3	4	4	X	-	-	-
B3-P1	19	4+19=23	X	-	-	-
B3-P2	100	4+19+100=123	-			
B3-P3	100	4+19+100=123	-			
B3-P4	100	4+19+100=123	-			
B4-P1	100	4+100=104	-			
B4-P2	4	4	X	-	-	-
B5-P1	4	4	X	-	-	-
B5-P2	19	4+19+100+19=142	-			
B6-P1	100	4+19+100=123	-			
B6-P2	19	4+19=23	X	-	-	-

Cálculo dos valores RPC das portas para determinar quais as portas *Root Port*

Portas Root Port



Portas *Root Port* assinaladas na rede.



Valores de Cálculo *Designated Port*

Porta	PC	RPC	RP	Segmento	DPC	DP	Block
B1-P1	4	-	-	S1	0	X	-
B1-P2	4	-	-	S2	0	X	-
B2-P1	100	104		S4	4	X	
B2-P2	19	142		S3	4	X	
B2-P3	4	4	X	S1	-	-	-
B3-P1	19	23	X	S3	-	-	-
B3-P2	100	123	-	S5	23	X	-
B3-P3	100	123	-	S5	23	-	X
B3-P4	100	123	-	S6	23	X	-
B4-P1	100	104	-	S4	4		X
B4-P2	4	4	X	S2	-	-	-
B5-P1	4	4	X	S2	-	-	-
B5-P2	19	142	-	S7	4	X	-
B6-P1	100	123	-	S6	23	-	X
B6-P2	19	23	X	S7	-	-	-

As portas que não *Root Port* nem *Designated Port* ficam no estado *Blocking*

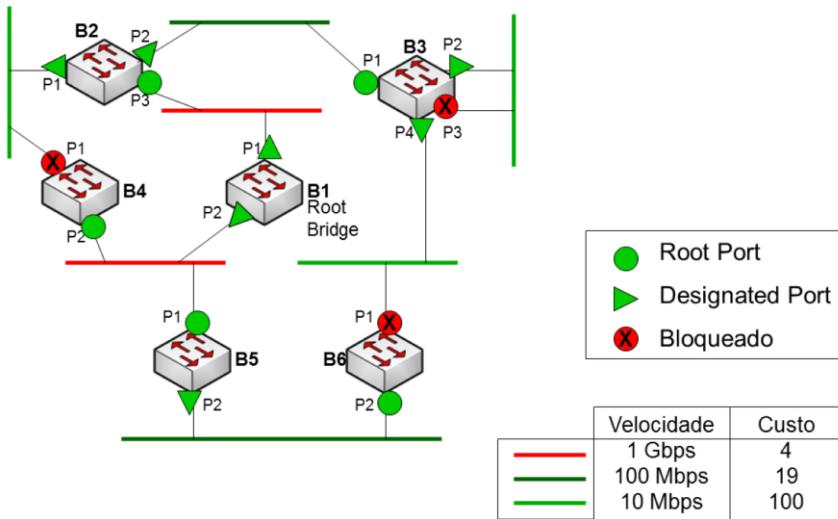
Cálculo dos valores *Designated Port Cost* das portas.

Cálculo das portas *Designated Port* de cada segmento.

Cálculo das portas *Blocked* (todas as que não são nem *Root Port* nem *Designated Port*)

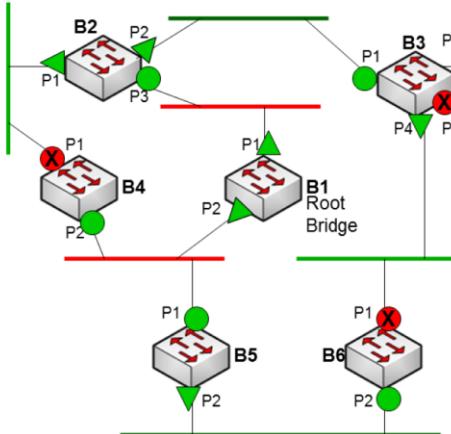
Atenção: Cada porta ativa pode ficar apenas num modo de funcionamento (*root*, *designated* ou no estado *blocking*).

Topologia Final da Rede

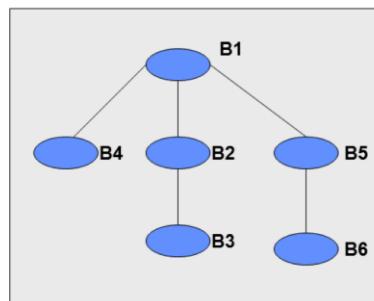


Topologia Final da Rede.

Topologia Final em Árvore Livre de Ciclos



Topologia da Rede em Árvore

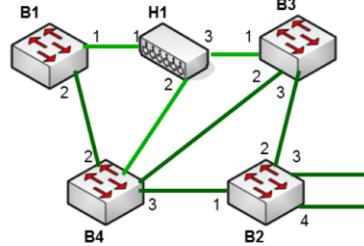


Verificação da existência de uma topologia de rede em árvore.

Exemplo Realizado em Laboratório



- Exercício prático para teste em laboratório
- 1º Os alunos fazem o cálculo da topologia de rede activa
- 2º Os alunos verificam experimentalmente os cálculos realizados e são confrontados com as diferenças
 - Dos identificadores das Bridges
 - Do protocolo RSTP em vez do STP



B1) 32768:00-0D-29-8F-00-01
B2) 30000:00-0D-29-8F-00-02
B3) 30000:00-0D-29-8F-00-03
B4) 32768:00-0D-29-8F-00-04

	Velocidade	Custo
—	100 Mbps	19
—	10 Mbps	100

Exercício prático para teste em laboratório

1º Os alunos fazem o cálculo da topologia de rede activa.

2º Os alunos verificam experimentalmente os cálculos realizados e são confrontados com as diferenças:

- Dos identificadores das Bridges – o que origina diferentes topologias. Por sorte pode calhar que a Root Bridge seja a mesma. Os alunos são encorajados a alterar as prioridades das Bridges e forçar uma topologia desejada.
- Do protocolo RSTP em vez do STP. Actualmente as Bridges implementam o protocolo RSTP o que provoca algumas alterações de valores.

Referências



- Norma IEEE 802.1D-1998
 - Norma IEEE 802.1D-2004
 - Fred Halsall, "Data Communications, Computer Networks", 4^a Edição
-

Referências usadas na lição.